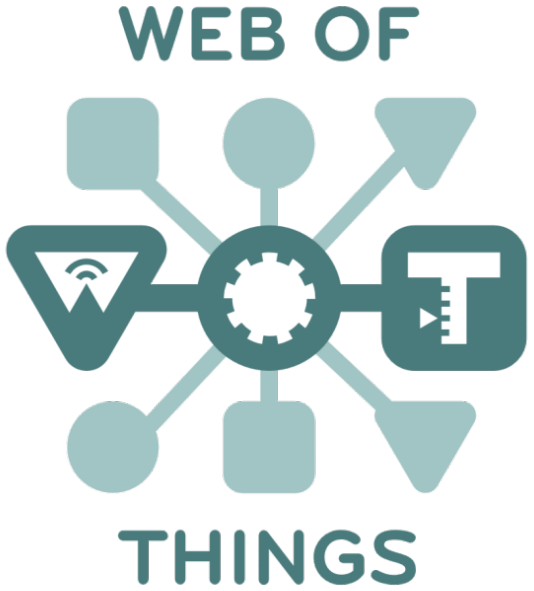


Decentralized Identifiers: Summary

Michael McCool, February 2020

Outline

- Goal: Summarize DIDs (Decentralized Identifiers)
- Simple Example: DID and DID Document
- DID Use Cases
  - Use Cases and Requirements for Decentralized Identifiers
  - Requirements
  - Design Goals
  - Key Terminology
  - Applicable Use Case: Accessing Service Endpoints
  - DID Actions
- DID Core Concepts
  - Decentralized Identifiers (DIDs) v1.0
  - DID URIs
  - DID Documents
- Possible Applicability to WoT
  - TDs
  - Discovery
- Other References

DIDs and DID Documents: Simple Example

`did:example:123456789abcdefghi`

… resolves to …

```
{
  "@context": "https://www.w3.org/ns/did/v1",
  "id": "did:example:123456789abcdefghi",
  "authentication": [{
    // used to authenticate as did:...fghi
    "id": "did:example:123456789abcdefghi#keys-1",
    "type": "RsaVerificationKey2018",
    "controller": "did:example:123456789abcdefghi",
    "publicKeyPem": "-----BEGIN PUBLIC KEY...END PUBLIC KEY-----\r\n"
  }],
  "service": [{
    // used to retrieve Verifiable Credentials associated with the DID
    "id":"did:example:123456789abcdefghi#vcs",
    "type": "VerifiableCredentialService",
    "serviceEndpoint": "https://example.com/vc/"
  }]
}
```

Basic Requirements

Decentralized:

- there should be no central issuing agency;

Persistent:

- the identifier should be inherently persistent, not requiring the continued operation of an underling organization;

Crytopgraphically verifiable:

- it should be possible to prove control of the identifier cryptographically;

Resolvable:

- it should be possible to discover metadata about the identifier.

Design Goals

Goal	Description
Decentralization	Eliminate the requirement for centralized authorities or single point failure in identifier management, including the registration of globally unique identifiers, public verification keys, service endpoints, and other metadata.
Control	Give entities, both human and non-human, the power to directly control their digital identifiers without the need to rely on external authorities.
Privacy	Enable entities to control the privacy of their information, including minimal, selective, and progressive disclosure of attributes or other data.
Security	Enable sufficient security for relying parties to depend on DID documents for their required level of assurance.
Proof-based	Enable DID subjects to provide cryptographic proof when interacting with other entities.
Discoverability	Make it possible for entities to discover DIDs for other entities to learn more about or interact with those entities.
Interoperability	Use interoperable standards so DID infrastructure can make use of existing tools and software libraries designed for interoperability.
Portability	Be system and network-independent and enable entities to use their digital identifiers with any system that supports DIDs and DID methods.
Simplicity	Favor a reduced set of simple features to make the technology easier to understand, implement, and deploy.
Extensibility	Where possible, enable extensibility provided it does not greatly hinder interoperability, portability, or simplicity.

Key Terminology

decentralized identifier (DID):

A globally unique identifier that does not require a centralized registration authority because it is registered with distributed ledger technology (DLT) or other form of decentralized network.

DID controller:

The entity, or a group of entities, in control of a DID or DID document. Note that the DID controller might include the DID subject.

DID document:

A set of data describing the DID subject, including mechanisms, such as public keys and pseudonymous biometrics, that the DID subject can use to authenticate itself and prove their association with the DID.
A DID document might also contain other attributes or claims describing the subject.
These documents are graph-based data structures that are typically expressed using [JSON-LD], but can be expressed using other compatible graph-based data formats.

DID method:

A definition of how a specific DID scheme can be implemented on a specific distributed ledger or network, including the precise methods by which DIDs are resolved and deactivated and DID documents are written and updated.

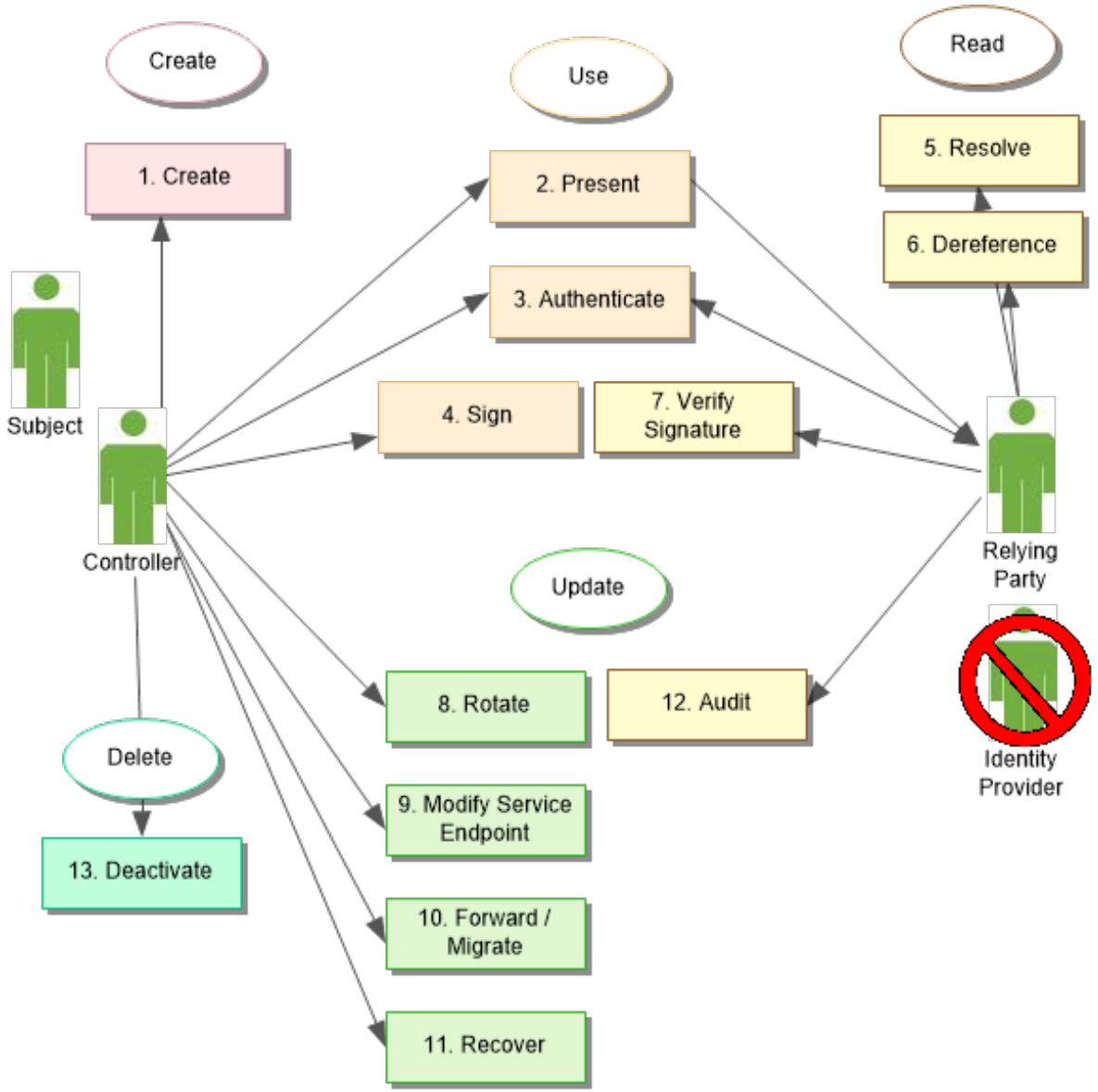
DID subject:

The entity the DID document is about.
That is, the entity identified by the DID and described by the DID document.

service endpoint:

A network address at which a service operates on behalf of a DID subject.
Examples of specific services include discovery services, social networks, file storage services, and verifiable claim repository services.
Service endpoints might also be provided by a generalized data interchange protocol, such as extensible data interchange.

DID Actions Related to CRUD Verbs



Applicable Use Case

2.4 Accessing service endpoints

- ***A Decentralized Identifier resolves to information relevant to that identifier including service endpoints – typically APIs that allow some sort of interaction with the identified item.*** For example, a service endpoint might provide corporate master data about the item which in turn could be provided as a self-signed verifiable credential. The decentralized nature of the identifier is important here as it can continue to act as an entry to that master data even if the manufacturer goes out of business.
- Another example would be parking spaces identified by a DID and connected to barriers. The static data directly associated with the parking space, such as its GPS position, can be made publicly available, perhaps leveraging schema.org, together with details of the secure service endpoint that facilitates interaction with the parking space's barrier. The parking space may be owned by the local authority but the barrier is likely to be operated by a contracted third party and that contracted party can be replaced at any time without needing to change the parking space's identifier.

DID URLs

- `did:method:identifier{;params}{/path}{#fragment}{?query}`
- Methods allow different kinds of DIDs to be defined
- Each method defines an identifier scheme and CRUD verb meanings
- Identifiers should be globally unique and immutable
  - Unique in the sense that there are no collisions
  - However, entities can have more than one identifier
  - Immutable in the sense that not changed, but they can be deactivated and a new one defined and associated with the entity
  - Some argumentation about whether or not to define DID equivalencies…
- There are both global and method-specific parameters allowed
- Paths can identify resources
- Fragments used to identify parts of a referenced DID document

DID Documents

- JSON-LD
  - With, apparently, some JSON-LD 1.1 features used…
  - In theory, other serialization schemes allowed
  - @context used to identify versions
  - Additional contexts can be defined, but examples do so in a scoped fashion
    - Can specifically be used to define new service types
- Structure
  - “id”: DID of the document
  - “updated” and “created” timestamps (in XSD datetime, UTC-Z format)
  - “controller”: DID of controlling entity (authorization delegation)
  - “publicKey”: array of public key data (each with id)
  - “authentication”: array of authentication metadata (each with id)
  - “service”: array of typed service endpoints (each with id)
  - “proof”: (optional) cryptographic proof of DID document integrity

Service Endpoint Examples

```
{ "service": [

    {
      "id": "did:example:123456789abcdefghi#openid",
      "type": "OpenIdConnectVersion1.0Service",
      "serviceEndpoint": "https://openid.example.com/"
    },

    {
      "id": "did:example:123456789abcdefghi#vcr",
      "type": "CredentialRepositoryService",
      "serviceEndpoint": "https://repository.example.com/service/8377464"
    },

    {
      "id": "did:example:123456789abcdefghi#xdi",
      "type": "XdiService",
      "serviceEndpoint": "https://xdi.example.com/8377464"
    },

    {
      "id": "did:example:123456789abcdefghi#agent",
      "type": "AgentService",
      "serviceEndpoint": "https://agent.example.com/8377464"
    },

    {
      "id": "did:example:123456789abcdefghi#hub",
      "type": "IdentityHub",
      "publicKey": "did:example:123456789abcdefghi#key-1",
      "serviceEndpoint": {
        "@context": "https://schema.identity.foundation/hub",
        "type": "UserHubEndpoint",
        "instances": ["did:example:456", "did:example:789"]
      }
    },

    {
      "id": "did:example:123456789abcdefghi#messages",
      "type": "MessagingService",
      "serviceEndpoint": "https://example.com/messages/8377464"
    },

    {
      "id": "did:example:123456789abcdefghi#inbox",
      "type": "SocialWebInboxService",
      "serviceEndpoint": "https://social.example.com/83hfh37dj",
      "description": "My public social inbox",
      "spamCost": {
        "amount": "0.50",
        "currency": "USD"
       }
    },

    {
      "id": "did:example:123456789abcdefghi#authpush",
      "type": "DidAuthPushModeVersion1",
      "serviceEndpoint":
      "http://auth.example.com/did:example:123456789abcdefg"
    }

    ]}
```

Possible Applicability to WoT

- Use of DIDs as Thing ids
- Question: What should the DID Document related to a Thing contain?
  - Things as service endpoints? – insufficient information on Thing interactions
  - Thing resources as service endpoints? – replicates TDs, with less structure
  - Thing Descriptions as service endpoints? – possibly appropriate
  - Thing Description Directories as service endpoints? – possibly appropriate
    - Perhaps with suitable query parameters to find a specific Thing Description
- Observation:
  - DID Documents' service lists are similar to CoRE RD data: lists of typed links
- Discovery:
  - We should explore and discuss discovery approaches based on DLTs and DIDs
  - Need to decide upon role of DID Documents wrt TDs
    - May have similar relationship as CoRE RDs do with TDs
- Security:
  - DID Documents can contain public key data; can we reference this in Thing Description security schemes?

Other References and Related Standards

- DID Resolution: https://w3c-ccg.github.io/did-resolution/
- DID WG Minutes: https://www.w3.org/2019/did-wg/Meetings/Minutes/
- Wiki: https://ldapwiki.com/wiki/W3C%20Decentralized%20Identifiers
- DID Primer: https://w3c-ccg.github.io/did-primer/
- DID WG landing page: https://www.w3.org/2019/did-wg/
- Implementation guide (skeleton): https://github.com/w3c/did-imp-guide
- Related standards:
  - IETF/TCG - Device ID - Implicit Identifier (note use of "DID" for Device ID… a naming conflict!)
  - DID-DNS: https://datatracker.ietf.org/doc/draft-mayrhofer-did-dns/
  - IoT EPI
- Possible overlaps with WoT Discovery – "resolution" capability:
  - https://w3c-ccg.github.io/did-method-registry/
  - Registry issues: https://github.com/w3c/did-use-cases/issues/14
- Relationship to TLS, PoP tokens, etc?
- Blogs and articles found:
  - https://medium.com/metadium/decentralized-identifiers-the-easy-guide-fb96429e8b24
  - https://medium.com/@adam_14796/understanding-decentralized-ids-dids-839798b91809
- Other references:
  - https://www.w3.org/TR/did-core/#references D. References (did-core)
- Older work (CCG):
  - https://w3c-ccg.github.io/
- Privacy by Design:
  - https://en.wikipedia.org/wiki/Privacy_by_design